

第一章

一、名词解释

1. 操作系统: 管理计算机软件 and 硬件资源的集合系统
2. 预输入: 事先将装有用户程序和数据的纸带装入输入机, 在一台外围机的控制下, 把纸带上的程序输入磁带。当 CPU 需要这些程序和数时, 再从磁带上高速调入内存。
3. 分布式操作系统: 配置在分布式系统上的公用操作系统。
4. 实时操作系统: 系统能及时响应外部事件的请求, 在规定的时间内完成对该事件的处理, 并控制所有实时任务协调一致地运行。
5. 互斥共享: 系统内的某些资源虽然可以提供给多个进程, 但应规定在一段时间内只允许一个进程访问该资源。即在共享资源时, 必须使用互斥方式。

三、填空

1. 一个完整的计算机系统包括硬件系统和软件系统。
2. 只有计算机硬件而没有任何软件支持的计算机称为裸机。
3. 计算机系统资源包括硬件资源和软件资源两部分。
4. 单道批处理系统的特点包括单道性、顺序性和自动性。
5. 操作系统的特征包括并发性、共享性、异步性和虚拟性。
6. 处理机管理主要包括进程控制、进程同步、进程通信和进程调度。

第二章

一、名词解释

3. 临界区: 在多个进程或线程中访问共享资源的代码段, 需互斥控制, 只保证一个进程/线程同时访问临界区。
4. 进程同步: 异步环境下的一组并发进程因直接制约而互相发送信息、互相合作、互相等待, 使得各进程按一定速度执行的过程。

三、填空

1. 动态性、并发性、独立性、异步性。
5. 动态, 静态
6. 间接制约
8. 共享存储器、管道通信、消息传递

9. 就绪状态、执行状态、阻塞状态

五. 问答题

2. 前驱关系: $S_1: \text{无}$, $S_2: S_1$, $S_3: S_1$, $S_4: S_2$, $S_5: S_2$, $S_6: S_3$, $S_7: S_4, S_5, S_6$.

信号量机制实现: semaphore $a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0$

```
P1() { S1; P2() { P(a); P3() { P(b); P4() { P(c); P5() { P(d);
    V(a); S2; S3; S4; S5;
    V(b); } V(c); V(e); } V(f); } V(g); }
```

```
P6() { P(e); P7() { P(f);
    S6; P(g);
    V(h); } P(h);
    S7; }
```

3. 盘子互斥访问, 以操作同步.

Semaphore $apple=0, orange=0, empty=5, mutex=1$;

```
Father() { Son() { Daughter() {
    while(1) { while(1) { while(1) {
        P(empty); P(orange); P(apple);
        P(mutex); P(mutex); P(mutex);
        放入苹果或桔子; 拿桔子; 拿苹果;
        V(mutex); V(mutex); V(mutex);
        V(apple); 或 V(orange); } } V(empty); } }
```

4. Semaphore $empty1=1, empty2=1, full1=0, full2=0$;

```
PA() { PB() { PC() {
    while(1) { while(1) { while(1) {
        P(empty1); P(full1); P(full2);
        读入缓冲区1; 复制到缓冲区2; 打印;
        V(full1); } V(empty1); V(empty2);
        P(empty2); 存入缓冲区2; } }
        V(full2); }
```

5. `int wait=0;`
`Semaphore customer=0; barber=0; mutex=1;`

```
Customer() {
    if (wait < N) {
        wait++;
        V(mutex);
        V(customer);
        P(barber);
    }
    else { V(mutex);
        P(customer); }
}

Barber() {
    P(customer);
    P(mutex);
    wait--;
    V(barber);
    V(mutex);
    V(customer);
}
```

第三章

一、名词解释:

1. 处理机调度: 操作系统根据特定的处理机算法, 从就绪队列中选择一个进程或线程并分配给它CPU资源执行的过程。
2. 周转时间: 作业从提交到OS到完成所经历的时间。
3. 死锁: 多个进程因竞争系统资源或相互等待对方占有的资源, 导致进程永远处于阻塞状态, 导致系统无法正常运行。

二、填空

2. 提交、后备、执行以及完成。
6. 预防、避免、检测、解除。

五、问答题

1. ① FCFS 先来去服务。

进程	A	B	C	D	E	Average	进程	A	B	C	D	E	Average
完成t.	3	9	13	18	20	8.6	完成t.	3	9	15	20	11	
周转t.	3	7	9	12	12	8.6	周转t.	3	7	11	14	3	7.6
带权周转t.	1	1.17	2.25	2.4	6.236	2.36	带权周转t.	1	1.17	2.75	2.8	1.5	1.84

② 非抢占SJF优先

③ 抢占SJF先。

进程	A	B	C	D	E	Average
完成t.	3	15	8	20	10	7.2
周转t.	3	13	4	14	2	
带权周转t.	1	2.16	1	2.8	1	1.59

④ HRRN 高响应比优先

进程	A	B	C	D	E	Average
完成 t	3	9	13	20	15	
周转 t	3	7	9	14	7	8
带权周转 t	1	1.27	2.25	2.8	3.5	2.14

$t=0$: A 先上.

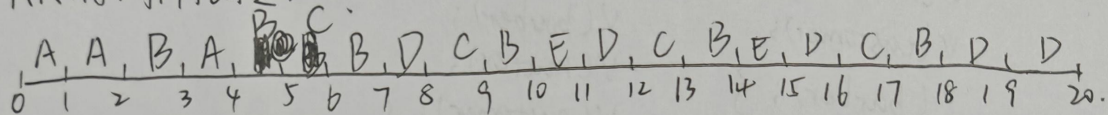
$t=3$: B 先上.

$t=9$: C 响应比最大, C 上.

$t=13$: E 响应比最大, E 上.

$t=15$: D 上.

⑤ RR 时间片轮转



进程	A	B	C	D	E	Average
完成 t	4	18	17	20	15	
周转 t	4	16	13	14	7	8
带权周转 t	1.33	2.67	3.25	2.8	3.5	2.14

2. 设任务进程同时到达, 松弛度 = 截止时间 - 运行时间 - 当前时间.

$t=0$ 时: A: $20-10-0=10$ B: $50-10-0=40$ C: $50-15-0=35$

A 先执行.

$t=10$ 时: B: $50-10-10=30$ C: $50-15-10=25$ C 先执行.

$t=25$ 时: A: $20 \times 2 - 10 - 25 = 5$ B: $50-10-25=15$ A 先执行.

$t=35$ 时: B: $50-10-35=5$ C: $50 \times 2 - 15 - 35 = 50$ B 先执行.

$t=45$ 时: B 执行完毕. 一轮结束.

3. (1) 分配 = MAX-Need.

进程	分配 Allocation
P0	0 0 3 2
P1	1 0 0 0
P2	1 3 5 4
P3	0 3 3 2
P4	0 0 1 4

(2) Request₂(1, 2, 2, 2) < Need₂(2, 3, 5, 6).

Request₂(1, 2, 2, 2) > Available₂(1, 6, 2, 2).

则: Available₂ = (0, 4, 0, 0), 不满足任何 Available >= Need. 不安全. 不能分配.

(3) 不会立即进入死锁.

该状态不安全.